We claim:

1. A multi-data-stream merge network comprising a pipelined butterfly network for receiving parallel streams of data which are in interleaved order and which are located at specific spatial boundaries, and which includes means for concatenating data from the received streams of data into constant width interleaved words having width which is wider than any of the received streams of data, and means for merging the constant width words onto an output bus in a time-division multiplexed form so as to produce a constant-width interleaved output data stream, wherein the pipelined butterfly network is comprised of $j*\log_2 j$ successive stages of multiplexers and $j*(\log2j+1)$ delay stages, in which j represents the number of parallel lines of byte lanes of input streams of data, the multiplexers being interconnected in a butterfly pattern, apparatus for applying select signals to the multiplexers which select signals toggle after 2^i clock signals for the i^{th} stage of multiplexers following a starting value for a sequence of select signals offset by k modulo$2^{(i-1)}$ for each successive multiplexer in the i^{th} stage of multiplexers.

2. A network as defined in claim 1, further comprising an input delay network containing P serial delay stages in which each respective successive parallel input data stream is delayed by said P serial delay stages, where P=(M-1), M being a whole number counting from 1 representing a count of each respective successive input data stream, said input delay network receiving the streams of data and passing each stream to an input of a corresponding multiplexer of a first stage of said multiplexers.

3. A network as defined in claim 1, further comprising an output network for providing an output data stream from a last stage of said multiplexers, the output network being comprised of an output delay network comprising a number of delay stages Q in each parallel stream carrying the output data stream wherein $Q=(R-X)$, where R represents the total number of parallel output lines carrying the output data stream and X represents a whole number counting from 1 representing a count of each successive line forming the output bus carrying the output data stream.

4. A network as defined in claim 2, further comprising an output network for providing an output data stream from a last stage of said multiplexers, the output network being comprised of an output delay network comprising a number of delay stages Q in each parallel stream carrying the output data stream wherein $Q=(R-X)$, where R represents the total number of parallel output lines carrying the output data stream and X represents a whole number counting from 1 representing a count of each successive line forming the output bus carrying the output data stream.

5. A network as defined in claim 4, further comprising a plurality of shuffle buffers for receiving plural input data streams

(a) each data stream but one having a width which is the same as all other input data streams, and

(b) at least one stream of data having different data width than other ones of the input streams of data and in which the data widths of the streams of data have a power

of 2 relationship;

the shuffle buffer having a structure for reordering the input streams of data into an interleaved order if not already in interleaved order and providing the interleaved order of data streams at its output.

6. A network as defined in claim 5, further including a permutation network located upstream of the pipelined butterfly network, for receiving streams of data and for rearranging the spatial order of the streams of data if desirable or necessary and locating each stream on a specific spatial boundary, wherein the permutation network is configured to reorder positions of data streams so that byte lanes of respective streams which are a multiple of the narrowest of the data streams are respectively contiguous and are located on a specific spatial boundary.

7. A network as defined in claim 6, wherein at least one of the input streams of data has a wider data width, K being the width of said wider data stream, than other ones of the input streams of data, and the sum of the input streams of data does not exceed the data width of the output bus, and wherein the first $\log_2 K$ stages of multiplexers of the pipelined butterfly network to which a data stream having the wider data width are applied are frozen so as not to toggle, with their select inputs set so as to render those stages of multiplexers into a straight-through data passage mode.

8. A network as defined in claim 5, in which all of the input data streams are of the same width, and further comprising buffers for at least one of

(a) accumulating input data until complete blocks of data are available within the buffers which are of a size equal to the width of the output bus of the pipelined butterfly network, and

(b) synchronizing the input data streams; and for outputting the at least one of the synchronized input data streams and blocks of data for subsequent processing by the pipelined butterfly network.

9. A network as defined in claim 8, further including a permutation network located upstream of the pipelined butterfly network, for receiving streams of data and for rearranging the spatial order of the streams of data if desirable or necessary and locating each stream on a specific spatial boundary.

10. A network as defined in claim 2, further comprising a plurality of shuffle buffers for receiving plural input data streams

(a) each data stream but one having a width which is the same as all other input data streams, and

(b) at least one stream of data having different data width than other ones of the input streams of data and in which the data widths of the streams of data have a power of 2 relationship;

the shuffle buffer having a structure for reordering the input streams of data into an interleaved order if not already in interleaved order and providing the interleaved order of data streams at its output.

11. A network as defined in claim 10, wherein at least one of the input streams of data has a wider data

width, K being the width of said wider data stream, than other ones of the input streams of data, and the sum of the input streams of data does not exceed the data width of the output bus, and wherein the first $\log_2 K$ stages of multiplexers of the pipelined butterfly network to which a data stream having the wider data width are applied are frozen so as not to toggle, with their select inputs set so as to render those stages of multiplexers into a straight-through data passage mode.

12. A network as defined in claim 11, further comprising an input delay network containing P serial delay stages in which each respective successive parallel input data stream is delayed by said P serial delay stages, where P=(M-1), M being a whole number counting from 1 representing a count of each respective successive input data stream, said input delay network receiving the streams of data and passing each stream to an input of a corresponding multiplexer of a first stage of said multiplexers.

13. A network as defined in claim 11, further comprising an output network for providing an output data stream from a last stage of said multiplexers, the output network being comprised of an output delay network comprising a number of delay stages Q in each parallel stream carrying the output data stream wherein Q=(R-X), where R represents the total number of parallel output lines carrying the output data stream and X represents a whole number counting from 1 representing a count of each successive line forming the output bus carrying the output data stream.

14. A network as defined in claim 13, further including a permutation network located upstream of the pipelined butterfly network, for receiving streams of data and for rearranging the spatial order of the streams of data if desirable or necessary and locating each stream on a specific spatial boundary, wherein the permutation network is configured to reorder positions of data streams so that byte lanes of respective streams which are a multiple of the narrowest of the data streams are contiguous and are located on a specific spatial boundary.

15. A network as defined in claim 1, further comprising a plurality of shuffle buffers for receiving plural input data streams

(a) each data stream but one having a width which is the same as all other input data streams, and

(b) at least one stream of data having different data width than other ones of the input streams of data and in which the data widths of the streams of data have a power of 2 relationship;

the shuffle buffer having a structure for reordering the input streams of data into an interleaved order if not already in interleaved order and providing the interleaved order of data streams at its output.

16. A network as defined in claim 15, further including a permutation network located upstream of the pipelined butterfly network, for receiving streams of data and for rearranging the spatial order of the streams of data if desirable or necessary and locating each stream on a specific spatial boundary, wherein the permutation network is configured to reorder positions of data streams so that

byte lanes of respective streams which are a multiple of a narrowest of the data streams are contiguous and are located on a specific spatial boundary.

17. A multi-data-stream merge network comprising:

(a) a plurality of shuffle buffers for receiving plural input data streams

(i) each data stream but one having a width which is the same as all other input data streams, and

(ii) at least one stream of data having different data width than other ones of the input streams of data and in which the data widths of the streams of data have a power of 2 relationship;

the shuffle buffer having a structure for reordering the input streams of data into an interleaved order if not already in interleaved order and providing the interleaved order of data streams at its output.

(b) a permutation network for receiving the interleaved order of data streams and rearranging the spatial order of the data streams if desirable or necessary and locating each stream on a specific spatial boundary, and

(c) a pipelined butterfly network for receiving the data streams from the permutation network and concatenating data from the received data streams into constant width interleaved words having width which is wider than any of the input streams of data, and merging the constant width words onto an output bus in a time-division-multiplexed manner.

18. A merging network for multiple data streams comprising a pipelined butterfly network, the pipelined butterfly network comprising:

(a) an input network for receiving a plurality of data streams of mutually constant widths, having logically related data streams carried on contiguous signal lines,

(b) a butterfly network for rearranging the received data streams into a time-multiplexed constant-width output data stream, the output data stream having a width equal to or greater than the sum of the widths of the input data streams, and

(c) an output network for providing the output data stream interleaved to an output bus.

19. A pipelined butterfly network as defined in claim 18, wherein the input network is comprised of an input delay network having a number P of equal delay stages, where P = (M-1), M being a whole number counting from 1 representing a count of each successive input data stream.

20. A pipelined butterfly network as defined in claim 18, wherein the output network is comprised of an output delay network having similar number of equal delay stages as the input delay network, each delay stage being equal in delay to a delay stage of the input network, the number of delay stages Q in each parallel line carrying the output data stream being Q = (R-X), where R represents the total number of parallel output lines carrying the output data stream and X represents a whole number counting from 1 representing a count of each successive output data stream.

21. A pipelined butterfly network as defined in claim 20, wherein the butterfly network is non-blocking and is comprised of a plurality of stages of multiplexers and delay elements mutually interconnected and selected so as to

provide to each parallel line containing said delay stages carrying the output data stream, an ordered and sequential stream of bytes from each successive word of all the input data streams, the sequence of bytes on successive ones of the parallel lines having successive bytes of a data word each delayed by a delay interval of one delay stage.

22. A pipelined butterfly network as defined in claim 21 further including a plurality of input buffers each for receiving a corresponding ordered and sequential stream of bytes, and for providing said corresponding stream of bytes to inputs of the butterfly network in accordance with a predetermined control signal and with a delay of one of said delay stages, and a control circuit for applying control signals to said input buffers and to said multiplexers.

23. A pipelined butterfly network a defined in claim 22 including a clock input to each of the multiplexers, buffers and delay stages for receiving a clock signal for synchronization thereof.

24. A pipelined butterfly network as defined in claim 21 comprising N x $\log_2 N$ multiplexers and N x ($\log_2 N$ + 1) delay elements, wherein N represents a total number of bytes in a data word at the output of the butterfly network.

25. A pipelined butterfly network as defined in claim 21 in which the received data streams are comprised of one of (a) all of the same width, or (b) different widths wherein the different widths are related to each other by the power of 2; the sum of the widths of the data streams

not exceeding the width of the output bus.

26. A pipelined butterfly network a defined in claim 22 including means for providing control signals, comprised of a regular repeating pattern.

27. A method of merging multiple input data streams into an output data stream having a word size that is equal to or larger than the sum of all of the input data streams, wherein the word size of the output data stream is a multiple of the smallest common factor of the word sizes of the input data streams, comprising:

(a) delaying each successive input data stream by a delay clock unit equal to (M-1), M being a whole number representing a count of each successive input data stream counting from 1,

(b) switching each successive byte of each delayed data stream to a corresponding successive internal output line and wherein each corresponding byte of all of the input data streams is in sequence on a corresponding internal data line, and

(c) delaying data streams on each of the internal data lines by respective delay clock units so as to cause all of the sequential bytes of an input data word to appear at the same time in parallel on respective output lines of an output bus, and all of the corresponding bytes of all data words to appear respectively at the same time in parallel, in sequence, and intereleaved, on the output bus.

28. A method as defined in claim 27 in which step (c) is comprised of delaying the data on each internal data line by a number of delay units equal to Q, wherein Q = (R-

X), where R represents the total number of parallel output lines carrying the output data stream, and X represents a whole number counting from 1 representing a count of successive output data lines carrying the output data stream.

29. A method of merging multiple input data streams into an output data stream having a word size that is equal to or larger than the sum of all of the input data streams, wherein the word size of the output data stream is a multiple of the smallest common factor of the word sizes of the input data streams, comprising:

(a) receiving a plurality of said input data streams having logically related data widths in an input network,

(b) rearranging the received data streams into a time-multiplexed constant-width output data stream, the output data stream having a width equal to or greater than the sum of the widths of the input data streams, and

(c) providing the output data stream interleaved to an output bus by means of an output network.

30. A method of merging multiple data streams into an interleaved time-division-multiplexed output signal comprising:

(a) receiving plural input streams of data, wherein one of (i) each stream of data has a data width which is the same as all other input streams of data, or

(ii) at least one stream of data has different data width than other ones of the input streams of data and in which the data widths have a power of 2 relationship,

(b) reordering the input streams of data into a

interleaved order if not already in interleaved order and providing the interleaved order of data streams at its output,

(c) rearranging the interleaved spatial order of the data streams if desirable or necessary and locating each stream on a specific spatial boundary, and

(d) concatenating data from the interleaved spatially located data streams into constant width interleaved words having width which is wider than any of the input streams of data, and merging the constant width words onto an output bus in a time-division-multiplexed manner.

31. A method of merging multiple data streams into an interleaved time-division-multiplexed output signal comprising receiving streams of data which are in interleaved order and which are located at specific spatial boundaries, concatenating data from the received streams of data into constant width interleaved words having a width which is equal to or is wider than any of the received streams of data, and merging the constant width words onto an output bus in a time-division multiplexed form so as to produce a constant-width interleaved output data stream.

32. A method as defined in claim 31 comprising imposing different fixed numbers of clock cycle delays on various input streams, switching and sequencing the delayed input streams in successive stages to merge and extend data in the delayed input streams, and imposing different fixed numbers of clock cycle delays on the merged and extended data so as to align the merged and extended data into interleaved words of the constant width output data stream.

33. A network for splitting a time-division multiplexed constant width interleaved data stream into multiple data streams including a pipelined butterfly network for receiving said interleaveddata stream, means for deconcatenating data from the received data stream into separate data streams and serializing each of the separate data streams into serial data streams of words having widths which are narrower than the width of the interleaved data stream, the widths of the serial data streams having a power of 2 relationship to each other.

34. A network as defined in claim 33, including a shuffle buffer for receiving the plural streams of data and for reordering the plural data streams in accordance with output data bus requirements.

35. A splitting network for converting an interleaved time-division multiplexed constant width data stream into multiple data streams comprising:

(a) an input network for receiving the interleaved time-division-multiplexed constant width parallel data stream,

(b) a butterfly network for rearranging the received data stream into a plurality of data streams having logically related widths and carried on contiguous signal lines, the sum of the widths of the plurality of data streams being equal to or smaller than the width of the interleaved data stream, and

(c) an output network for providing the plurality of data streams on plural output buses individually timed to output bus requirements.

36. A method of converting an interleaved time-division multiplexed constant width data stream into multiple data streams comprising:

(a) receiving the interleaved time-division-multiplexed constant width parallel data stream,

(b) rearranging the received data stream into a plurality of data streams having logically related widths and carried on contiguous signal lines, the sum of the widths of the plurality of data streams being equal to or smaller than the width of the interleaved data stream, and

(c) providing the plurality of data streams on plural output buses individually timed to output bus requirements.